Multi-modal Active Perception for Autonomously Selecting Landing Sites on Icy Moons

A. Arora[*] , P. M. Furlong[†] , U. Wong[†] , T. Fong[†] and S. Sukkarieh[*]

Selecting suitable landing sites is fundamental to achieving many mission objectives in planetary robotic lander missions. However, due to sensing limitations, landing sites which are both safe and scientifically valuable often cannot be determined reliably from orbit, particularly, in icy moon missions where orbital sensing data is noisy and incomplete. This paper presents an active perception approach to Entry Descent and Landing (EDL) which enables the lander to autonomously plan informative descent trajectories, acquire high quality sensing data during descent and exploit this additional information to select higher utility landing sites. Our approach consists of two components: probabilistic modeling of landing site features and approximate trajectory planning using a sampling based planner. The proposed framework allows the lander to plan long horizons paths and remain robust to noisy data. Results in simulated environments show large performance improvements over alternative approaches and show promise that our approach has strong potential to improve science return of not only icy moon missions but EDL systems in general.

I. Introduction

Icy moons such as Europa and Enceladus are among the top priorities for NASA's exploration objectives. These bodies may be the best candidates for finding life in the solar system, as interior liquid oceans may be present and accessible from the frozen surface. NASA has begun planning for a robotic lander mission to occur as soon as 2030.[1]

Due to the remote nature of these missions, human intervention is only possible at the most strategic levels and onboard autonomy is essential to safely execute complex maneuvers such as entry, descent and landing (EDL). The Curiosity Rover had, during its landing process, the "seven minutes of terror". Because of the communications delay between Mars and Earth there was a seven minute period between when the EDL procedure was initiated and when the operators on Earth would be aware of whether or not the rover successfully landed. In such EDL missions, the landing site is selected *a-priori* by domain experts based on orbital data, and onboard autonomy is limited to low level navigation, control and hazard avoidance. However, since the orbital data is often noisy, incomplete, and low resolution, the selected landing site might not be suitable for the science goals of the missions. This is especially the case in icy moon missions where key geological features such as crevasses, jagged penitentes, liquids and ice thickness may not be visible from orbit (Fig. 1).

This paper presents an active perception approach to autonomously select landing sites. The key idea is to plan informative descent trajectories such that the lander can acquire high quality observations of geological features and exploit this additional information to select landing sites which are both safe and have high science utilities. Our proposed approach consists of two components: probabilistic modeling of landing site parameters and informative trajectory planning.

We model landing site utilities as a Bayesian network (BN) which allows us to combine noisy data from different spatial scales and sensing modalities, and probabilistically estimate the safety and science utility of landing sites in a recursive manner. These estimates are then used to plan informative trajectories which direct the lander towards promising landing sites. We do this by adapting the Monte Carlo Tree Search (MCTS) algorithm which enables the lander to generate long horizon plans in an anytime manner while remaining robust to uncertainty.

[*] Australian Center for Field Robotics, The University of Sydney, Australia

[†] Intelligent Robotics Group, NASA Ames Research Center, California, USA

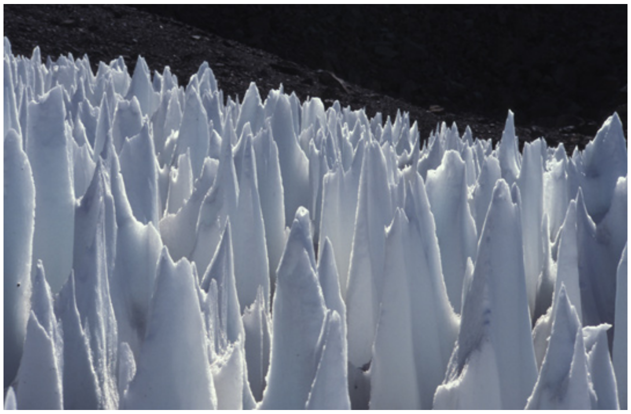

Figure 1. Penitente fields are jagged icy features theorized to exist on icy moons. These require low-altitude lander sensing to resolve and could be both hazards and features of science interest. Photo Credit: user:Arvaki via Wikimedia Commons/GFDL.

The main contributions of this paper are a formulation of the active EDL problem, an initial solution algorithm based on BNs and MCTS, and evaluation of the approach against existing techniques in simulated environments. We use an icy moon landing mission as a use case, but our approach is applicable to EDL missions in general.

II. Related Work

Traditional planetary EDL approaches have focused on hazard detection using computer vision and navigation techniques to accurately land in a desired location. The principle function of perception is to match low-altitude terrain to maps created from orbital data and to use relative motion to determine if a lander is excuting a predetermined trajectory[7].[8] Last minute diversions are allowed to avoid hazardous sites if detected, but not to tour multiple candidates and learn more about the environment scientifically. Bayesian frameworks have been proposed for planetary landing site selection to fuse reward estimates from multiple sensor sources, given all available information *a priori*.[5] Sensor sources are agnostic such that both geometric hazards and science utility can be represented. However, this work does not incorporate active exploration or online learning of the environment to update beliefs as the mission is flown. The work of Desaraju, et al., which explores terrestrial rooftop environments with a UAV to select the best landing site, uses a Gaussian Process approach and is similar to the idea presented in this paper.[4]

However, our approach extends prior work by considering an online component so that landers can acquire the most important data while exploring, reason about fuel constraints, incorporate scheduling costs of using disparate sensors, and consider the increasing quality of data as the spacecraft is closer to the ground. The principal contribution of this work is to integrate temporal information in the EDL decision making process. The flexibility of our framework enables landers to be active explorers and to respond to unknowns present on remote, icy moons where rigid pre-programming would fail.

We also consider the science utility of the site rather than just geometry. While typical EDL sensor packages have used lidar and cameras to characterise geometry, we propose to use non-traditional sensors like bore-sight imagers, thermal cameras, and sounding radar which can augment geometry with better science utility cues (bio-markers, etc).

For planning trajectories, we use Monte Carlo Tree Search (MCTS) methods which are sampling based, approximate tree search algorithms.[10] They have an advantage over both gradient descent and other sampling based approaches like Rapidly Exploring Random Trees (RRTs) by being anytime, allow reasoning over long horizons quickly, and easier to apply to situations where the environment is partially observable.[11] These properties are particularly advantageous in an EDL situation where there are hard real time constraints and observations from sensors are noisy and incomplete.

III. Problem Formulation

This section describes the properties of the lander, how we model the environment and formally defines the planning problem that needs to be solved to generate informative trajectories which select good landing sites.

Environment Representation: We use a grid world representation of the environment where each grid cell is a potential landing site. Each cell is described by some feature vector F. In icy moon environments, these features could include ice thickness, terrain jaggedness, slope and thermal properties. We assume that there are functions available that map the feature vector to the scientific value, $V_S : F \rightarrow [0, \ldots, \infty)$, and the trafficability, or safety, $V_T : F \rightarrow \{Unsafe, Safe\}$, of the candidate landing sites. The overall site utility U is then defined as some function of the scientific value and the safety probability of the site.

In practice, due to sensing limitations and energy constraints, the lander cannot deduce with 100% accuracy the true values of the features of all the sites. Therefore, a probability distribution over the true value of the feature vector along with the safety, science and overall utility is initialized from orbital data and refined during descent as observations are taken by the sensors on-board the lander vehicle.

Lander Properties: At any given time, the lander can choose to take a maneuvering action to change the direction of motion of the vehicle and select which one of its P-many sensors to use. An example payload for an icy moon lander could include a high resolution camera, ground penetrating radar, thermal sensor and a reflectance spectrometer. Each sensor observes different subsets of the feature vector F and has its own noise model and field of view which varies with the spacecraft altitude. We discretize the maneuvering space m into K actions. This produces an action space, $A = \{m_1, \ldots, m_K\} \times \{s_0, s_1, \ldots, s_P\}$.

We can formalize the active perception problem as follows: The lander must plan a sequence of maneuvering and sensing actions $a_{1\ldots L}$ which maximize some reward function R measuring the likelihood of landing at a site with high overall utility. Each maneuvering or sensing action a_i incurs some predefined cost given by the $cost(a_i)$ function and the overall sequence is subject to some general budget B. This budget could be the delta-V or time. The optimization problem is stated below:

$$\begin{aligned} a^*_{1..L} &= \underset{a_{1\ldots L} \in A}{\arg\max}\, R(a_{1\ldots L}) \\ \textbf{s.t.} \sum_{i=1}^{L} &\mathrm{cost}(a_i) \leq B \end{aligned} \tag{1}$$

The optimal sensing action sequence in one which in expectation will terminate at the best landing site with the highest probability. The reward function $R(\cdot)$ is therefore defined as:

$$R(a_{1\ldots L}) = \sum_{Z_{1\ldots L}} P(x_{best} = x_{chosen} | Z_{1\ldots L}) P(Z_{1\ldots L} | a_{1\ldots L}) \tag{2}$$

$Z_{1\ldots L}$ are the observations made in the sensing sequence, $P(Z_{1\ldots L} | a_{1\ldots L})$ is the sensor model while $P(x_{best} = x_{chosen} | Z_{1\ldots L})$ is a mapping of the observations made by the robot to the probability that the selected landing site has the highest utility. x_{chosen} is the landing site selected by the lander. The terminal location of the lander x_{final} must be within some radius R of the chosen landing site. This is given by the constraint below and we call this radius, the landing radius.

$$|x_{chosen} - x_{final}| \leq R \tag{3}$$

We now discuss the two main components of our approach: evaluating and updating candidate sites using Bayesian networks and planning informative descent trajectories using Monte Carlo Tree Search methods.

IV. Evaluating Candidate Sites

As mentioned in Sec. III, the environment is discretized into cells where each cell is a potential landing site and has a prior distribution over its geological features and utilities based on orbital images or previous measurements. The lander updates these distributions as observations are collected from on-board sensors during descent.

Evaluating Eq. 2 requires a mapping from on-board observations to the overall utility of a site. We use a Bayesian network (BN) to achieve this, similar to the approach of Serrano.[5] BNs are directed graphical

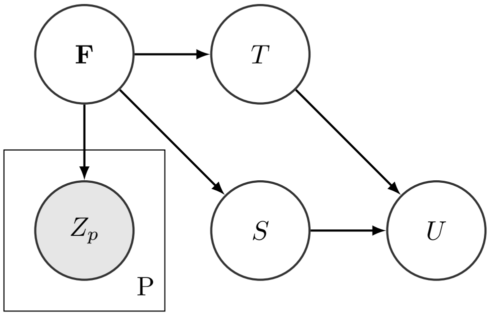

Figure 2. Bayesian network to calculate science utilities based on observations. Observations from the different senors, $Z_1, \ldots, Z_p$ inform the feature vector F in the grid cell. The feature vector informs the science value, S, and the safety of the terrain T.

models which describe causal dependencies and probabilistic relationships between variables. They are particularly attractive frameworks in this application because they give a principled approach for combining observations from different sources to make probabilistic inferences about the unobserved variables. We refer the reader to Nielsen and Finn[13] for an overview on BNs.

The BN we use is shown in Figure 2. The geological feature vector $\mathbf{F}$ is inferred by P on-board sensors through observations Z_p, where p represents the sensor used. Each sensor measures different subsets of these geological features from which the safety T, science utility S and overall utility U of a landing site can be estimated. In this problem setting we set the Z and $\mathbf{F}$ nodes to be discrete categorical variables as it simplifies inference, S as a continuous, non-negative, variable and T as a variable ranging from 0 to 1 indicating the probability whether a site if safe or not. There is an independent BN associated with each candidate landing site.

The conditional probability parameters of the BN quantify the probabilistic relationships between variables. It can be deduced that $P(Z_p|\mathbf{F})$ is the sensor model, while the $P(T|\mathbf{F})$ and $P(S|\mathbf{F})$ terms classify the safety and scientific utility of the site based on the geological features. We now define each of these terms in more detail.

Quantifying Landing Site Safety: The landing site needs to be classified as either "Safe" or "Unsafe", with a degree of belief in that classification. The $P(T|\mathbf{F})$ term is dependent on the actual features used as well as the design of the lander. It can be derived using domain knowledge as well as learning and classification techniques introduced previously in literature.[6,9] We assume this term is provided *a priori.*

Quantifying Landing Site Science Utility: Prior to the mission, scientists typically define the attributes they want in an ideal landing site in the form of a value function that maps features of a region to some score. In an icy moon mission, the features of interest may include presence of bio-markers, proximity to liquids, or desirable thermal properties of ice. We assume that the function which maps the geological features to a science utility value is known to the lander *a priori.* In this paper, we use a weighted linear function of geological features but any arbitrary function can be used. The science utility S of a landing site x is given by Eq. 4.

$$S(x) = \sum_{i}^{N} w_i F_i \tag{4}$$

Sensor Model: We now discuss how the sensor model term $P(Z|\mathbf{F})$ is calculated. The lander is equipped with several sensors. Each of these sensors has a rectangular field of view similar to that in Figure 3. The size of this footprint decreases with altitude, which in turn improves resolution and reduces sensor noise. The sensor model for sensor p at altitude a is given by Eq. 5 where $RMax$ is the maximum sensing range, $P(Z|\mathbf{F})_{best(p)}$ is the best case sensor noise model and G_p is the distribution representing the intrinsic noise model of the sensor. For example, a thermal camera may have a G_p that models pink noise while a laser altimeter would have a noise model that follows a uniform distribution.

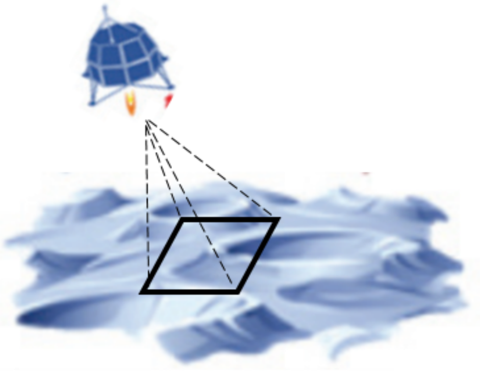

Figure 3. A typical field of view for a lander looking at the terrain.

$$
\begin{aligned}
P(Z|F)_{p,a} &= \alpha P(Z|F)_{best(p)} + (1-\alpha)G_p \\
\alpha &= \begin{cases} 0, & \text{if } a \geq RMax(p) \\ 1 - \frac{a}{RMax(p)}, & 0 \leq a < RMax(p) \\ 1, & a < 0 \end{cases}
\end{aligned}
\tag{5}
$$

The maximum range $RMax$, the $P(Z|\mathbf{F})_{best(p)}$ term, the intrinsic noise model and the type of features seen depend on the type of sensor used and assumed to be known *a-priori*. Sensor measurements taken throughout the mission are fed into the BNs to recursively update the safety and science utility estimates of each site along with the uncertainty using Bayes Theorem.

V. Planning Informative Descent Trajectories

With the ability to update the utility of candidate landing sites, the lander must plan sequences of actions that allow the lander to determine where the high utility landing sites are with high confidence and ensure the site can be reached given the sensing budget and the robot dynamics.

The optimization objective introduced in Equations 1 and 2 will allow optimal action sequences to be calculated, but requires estimating and summing over all possible observations that can be made. If the initial uncertainty in the observation space is high, running this calculation under the real time constraints of a landing is impractical. Furthermore, the space of trajectories the lander can choose from grows exponentially with the planning horizon. To address these problems we adapt a sampling based planning algorithm called Monte Carlo Tree Search (MCTS)[10] to plan future plans.

MCTS is a best first, anytime tree search algorithm which involves cycling through four stages: node selection, expansion, simulation and back-propagation. The key idea is to first select promising leaf nodes based on a tree policy. The selected node is expanded and a terminal reward is estimated by simulating future actions until the terminal state is reached. The reward is then back propagated up the tree and the process is repeated until some computational time limit is reached. At the end of the search, the child of the root node with the highest average reward is selected as the next best action. MCTS methods have made a significant impact in AI, particularly in stochastic games which have both long horizons and elements of uncertainty present in our problem.[10] Furthermore, MCTS is an anytime algorithm which means planning can be interrupted at any time and the current best action will be returned. This makes it particularly suitable for applications with hard real-time constraints.

In a typical EDL situation, the lander can control its 6DoF pose using its thrusters and have access to an arbitrary number of sensors. However, in this paper we illustrate the key ideas with a simplified version of the planning problem by making the following assumptions:

- The speed of the lander remains constant for the duration of the mission
- The altitude of the lander follows a predefined descent profile.

Algorithm 1 Descent Trajectory Planning for Selecting Landing Sites

```
**Input:** sensing budget $S$, belief maps of landing site utilities $B$, landing site evaluation BN $N$
**function** MAIN
    $R \leftarrow S$                                          ▷ R is the remaining budget
    **while** $R > 0$ **do**
        $landerPose \leftarrow getLocalisation()$             ▷ Can use arbitrary approximate localisation techniques
        $a_{opt} \leftarrow MCTSplanner(landerPose, R, B, N)$
        $Z \leftarrow takeObservation(a_{opt})$               ▷ Get an observation from the environment
        $B \leftarrow updateSiteUtilities(Z, B, N)$           ▷ Propagate new observations through the BNs
        $R \leftarrow R - cost(a_{opt})$                      ▷ Update remaining budget

**function** MCTSPLANNER($landerPose, R, B, N$)
    $T \leftarrow initialiseTree(landerPose, R)$              ▷ Create a tree
    $currentNode \leftarrow T.rootNode$                       ▷ Begin the MCTS at the root node
    **while** within computational budget **do**              ▷ Some time limit allocated to planning
        $[currentNode, treeSeq] \leftarrow UCT(T)$            ▷ Selects a leaf node with unexpanded children
        $simSeq \leftarrow simulationPolicy(currentNode, R, B)$
        $totSeq \leftarrow treeSeq + simSeq$                  ▷ Creates a path from root node to terminal state
        $reward \leftarrow getReward(landerPose, totSequence, B, N)$   ▷ Evaluate path reward
        $T \leftarrow updateTree(T, reward)$
    **return** $bestChild(T.rootNode)$                        ▷ Selects the action with the highest average reward

**function** SIMULATIONPOLICY($currentNode, R, B$)            ▷ Need to update this
    $simSeq \leftarrow 0$                                     ▷ Initialize action sequence vector
    **while** $R > 0$ **do**
        $aSpace \leftarrow getActions(currentNode, R)$        ▷ Get action space
        $a_i \leftarrow sample(aSpace)$                       ▷ Randomly choose an action
        $simSeq \leftarrow simSeq + a_i$                      ▷ Add new action to sensing sequence
        $R \leftarrow R - cost(a_i)$                          ▷ Update remaining budget
        $currentNode \leftarrow a_i(currentNode)$             ▷ Apply forward kinematics to get new spacecraft pose
    **return** $simSeq$

**function** GETREWARD($landerPose, totSequence, B, N$)
    **for** $i = 1 : length(totSequence)$ **do**
        $currentAction \leftarrow totSequence(i)$
        $Z \leftarrow sampleObs(currentAction, B)$            ▷ Sample an observation based on the current beliefs
        $B \leftarrow updateSiteUtilities(Z, B, N)$           ▷ Propagate observations through the BNs
    $landingSite \leftarrow getLandingSite(landerPose, totSequence)$   ▷ Get best landing site for trajectory
    $reward \leftarrow P(landingSite = bestSite|B)$           ▷ Probability that the landing site is the best site
    **return** $reward$
```

Under these assumptions, the lander is now restricted to only plan maneuvers in the x-y plane and decide which sensor to use. It is important to note that our planner does not require these assumptions to be true, they are only used to more clearly illustrate the methodology. The planner only requires a forward dynamic simulator to approximately predict future states given an action sequence.

We frame the descent trajectory planning problem as a decision tree where each node of the tree is a tuple consisting of the x-y position of the lander, orientation, altitude, velocity, the remaining budget and a variable indicating which sensor was used. The branches connecting the nodes are the actions the lander can take. The overall planning pipeline is described in Algorithm 1. We now discuss each of the four stages of MCTS in detail and how they have been used for our problem.

Selection and Expansion: The first stage of MCTS is selecting which leaf nodes to expand in the tree. We want to expand nodes which are expected to have a good terminal reward but at the same time evaluate alternative nodes enough to reduce chances of converging to local minima. There is an element of

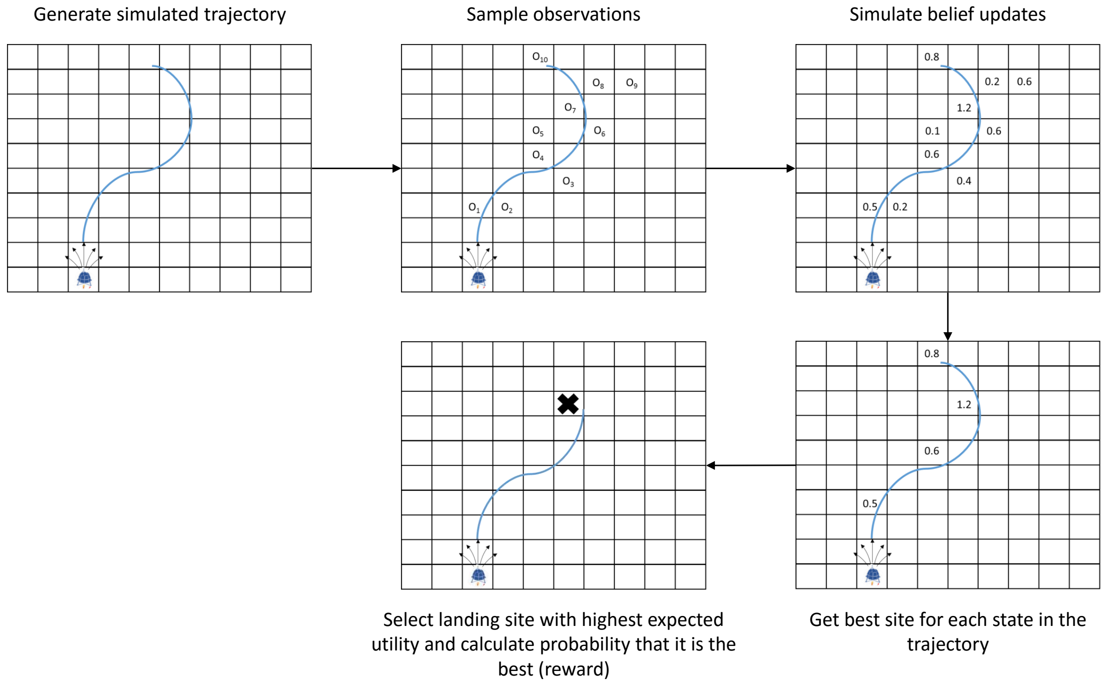
Generate simulated trajectory
Sample observations
Simulate belief updates
O_{10}
O_8
O_9
O_7
O_5
O_6
O_4
O_3
O_1
O_2
0.8
0.2
0.6
1.2
0.1
0.6
0.6
0.4
0.5
0.2
0.8
1.2
0.6
0.5
Select landing site with highest expected utility and calculate probability that it is the best (reward)
Get best site for each state in the trajectory

Figure 4. The 5 stage process for evaluating the reward of a simulated trajectory. The numbers in the top right plot are expectations on the total utility of landing sites.

exploration and exploitation present here. In this paper we use the Upper Confidence Tree (UCT) policy which is an effective approach in MCTS literature to deal with this dilemma.[12] The UCT policy is used until a leaf node with unexpanded children is reached. An unexpanded child is then randomly selected and added to the tree, this is the expansion step.

Simulation and Back-propagation: Next the MCTS assigns a reward to this newly added node by using some default simulation policy to guide the agent from the selected node to a terminal state. The reward of the total trajectory (both within the tree and the simulation phase) is evaluated and the average rewards of all nodes involved in the trajectory are updated by back propagation. When there is no prior knowledge available on where the high reward regions in the decision space lie, it is common to use a random policy for the simulations. However, many iterations of the MCTS are often required to adequately estimate future rewards. In this paper we develop our own simulation policy illustrated in Fig. 4.

Our simulation policy works as follows:

- A random action selection strategy is used from the current node until a terminal state is reached. A terminal state is either when the altitude of the lander reaches 0, the sensing budget is exhausted or any further actions will cause the lander to exit the map of possible landing sites.
- We then begin from the initial node in the trajectory, reason about sensor field of view at the current altitude to deduce what landing sites will be seen and sample observations for each site based on the current belief of feature space and the sensor noise at the current altitude.
- The belief space is updated based on the sampled observation and the process is repeated until the last state in the trajectory is reached. The overall utilities of all the sites is estimated based on the final beliefs of the features in each landing sites.
- For each state in the trajectory, the landing site with the highest expected overall utility within the landing radius R is determined.

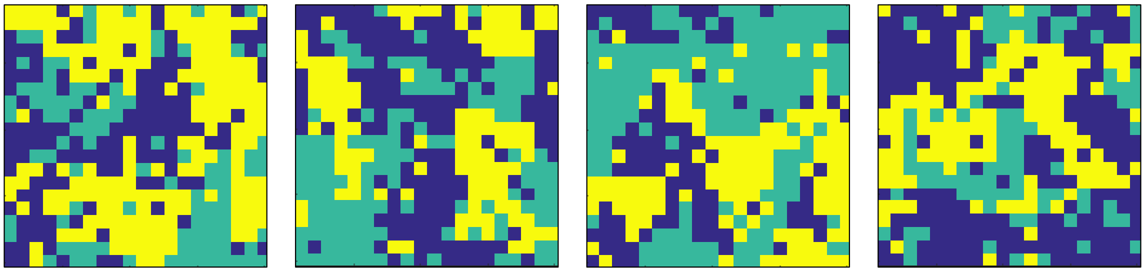

Figure 5. The feature grids used to characterize the 20×20 environment

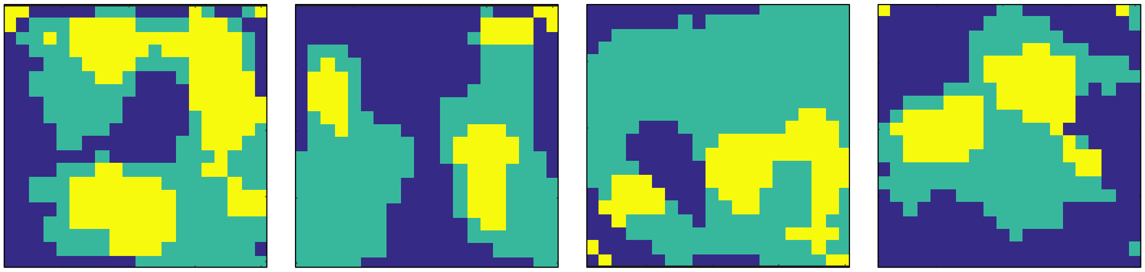

Figure 6. The orbital maps used to initialize the Bayesian priors on the features in each grid.

- The trajectory is now pruned such that it terminates at the best landing site encountered in the previous step.
- The reward is defined as the probability that the selected landing site has the highest overall utility out of the top 20 sites on the map with the highest expected utility. In our implementation, we fit Gaussian distributions to the utility estimate of each landing site and do pairwise comparisons to determine this probability.

The simulation policy is a design choice. Traveling salesmen heuristics and other methods such as potential field can also be used here but these require more computational time that our strategy. It has been shown in literature that MCTS generally performs better with large number of approximate simulations rather than a small number of accurate simulations.[14] Adding stochasticity to the policy is also fundamental to ensure the decision space is adequately explored. Our simulation policy achieves both of these requirements.

At the end of the computational budget assigned to planning, the child of the root node with the highest average reward is selected as the best action. Intuitively, actions which in expectation lead to landing sites with highest probabilities of being the best site will have the highest reward. This action is executed, an observation of landing sites is collected, belief maps are updated and the process is repeated with the updated lander position and budget until the landing is complete.

VI. Analysis

This section discusses the simulation settings used and results illustrating the effectiveness of our active perception approach over alternative approaches.

VI.A. Simulation Environment Setup:

Two environments of size 10×10 and 20×20 were generated where each grid cell is a potential landing site. The simulated environments were characterized by four arbitrary features labeled F_1, F_2, F_3 and F_4 which were discretized into three classes of low, medium and high.

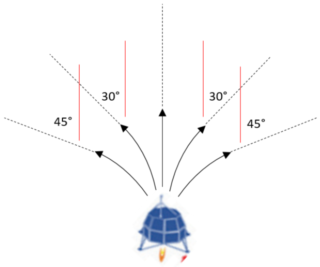

Figure 7. Motion primitives for the lander

These feature grids were generated by randomly labeling a subset of the cells in the grids and using these cells as seeds to grow voronoi regions. Random noise was then added to the feature grids to increase the spatial diversity of features within each voronoi region. The feature grids used for the 20×20 environment are shown in Fig. 5. In planetary bodies, landing sites close to each other are likely to have similar geographical features and it can be seen from the examples that the generated maps capture this relationship.

VI.B. Safety and science utilities:

In the experiments, it is assumed that the safety of the landing site is a function of features F_1 and F_2. This mapping is given by the matrix shown in Table 1. The science utility S of a landing site x is a weighted function of the labels of all four features in the corresponding cell. The function we use is given by Eq. 6 where the low, medium and high categories are assigned values of 1, 2 and 3 respectively. The overall utility is defined as the product of science and safety of a site. As observations are taken throughout the mission, the belief on safety, science utilities and overall utilities of sites is tracked and updated through particle filters.

Table 1. The mapping from feature space to the likelihood of the landing site being safety

		Feature 2		
		L	M	H
Feature 1	L	1	0.8	0.4
	M	0.8	0.6	0.2
	H	0.4	0.2	0

$$S(x) = 0.1F_{X1} + 0.2F_{X2} + 0.3F_{X3} + 0.5F_{X4} \tag{6}$$

VI.C. Lander Parameters

The simulated lander is equipped with two sensors: a visual sensor which can take noisy observations of features 1 and 2 and a spectrometer which can take noisy measurements of features 3 and 4. Both sensors have noise models discussed earlier in Eq. 5 and the parameters used are shown in Table 2. Both sensors also have a circular field of view with a viewing cone angle of 4 degrees.

Table 2. Sensor model parameters used in simulations

	Visual Sensor	Spectrometer
Maximum sensing range ($RMax$)	100	100
Intrinsic noise (G_p)	Uniform	Uniform
Maximum accuracy ($P(Z\|F)_{best}$)	80%	95%

As mentioned earlier, for illustration we simplified the planning problem to the x-y domain. The lander motion primitives were chosen to be Dubin's curves which orientate the lander in -45, -30, 0, 30 and 45 degrees

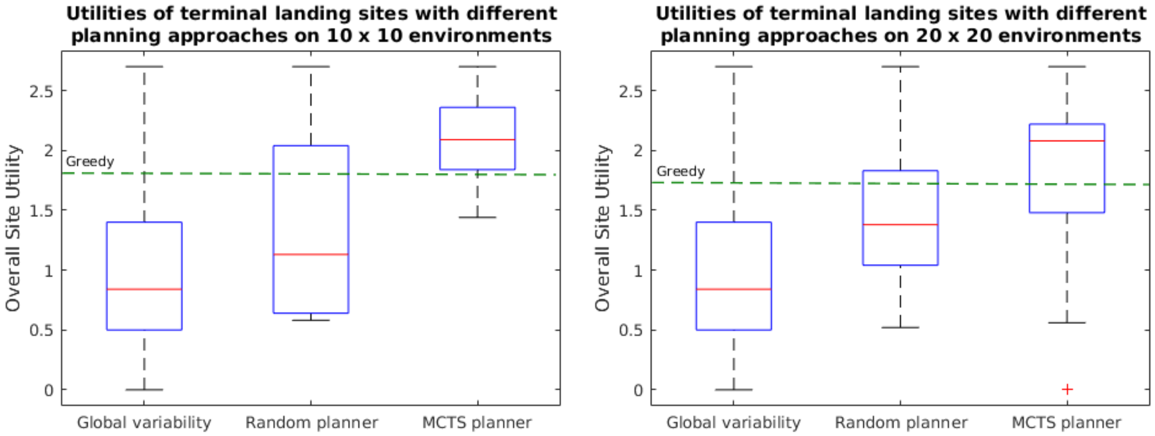

Figure 8. Results

(Fig. 7). Since there were two sensors, and in each time step the lander can choose a motion primitive and type of sensor to use, the total action space is of size 10. The lander velocity and descent rate is fixed for the duration of the simulation. The cost for using the visual sensor was 1 unit while the cost for using the spectrometer was 5 units.

VI.D. Orbital Maps

In EDL missions, it is common practice to create an orbital map of the planetary surface prior to landing. These maps help to give an approximate indication of where good landing sites may lie but due to limited sensing resolution miss finer geological features. Icy moons often have high surface reflectivity and contain liquids which make visual observations from orbit particularly noisy.

In this paper, the orbital maps were created by applying a median filter on the ground truth feature maps. The four orbital maps corresponding to the four features used in the 20×20 grid are shown in Fig. 6. It can be seen that much of the higher frequency variations in the feature maps as well as region boundaries have been smoothed out. This orbital data was encoded into landing site beliefs in the form of Bayesian priors on the feature space.

VI.E. Results

We compare the performance of the following approaches:

- A greedy policy which selects the best site as seen in the orbital map and lands there. This is analogous to a typical EDL approach where human operators choose a landing site based on orbital data, domain knowledge and science goals of the mission.
- A random action selection policy which takes a sequence of random actions until a terminal state is reached. Observations are collected during descent and probabilistic estimates of landing site properties are tracked. At the terminal state, the lander chooses the best site within its belief and within the landing radius as the landing site. This policy is an example of passive perception.
- Our approach, the MCTS active perception policy which like the random policy tracks probabilities of site utilities during descent but also actively plans sensing actions to direct the lander towards more promising sites.

20 trials were run for both random and MCTS policy on both 10×10 and 20×20 environments. The lander was initialized at an altitude and sensing budget of 50 units, while the velocity and descent rate were set to be 2 and 1 unit per time step respectively. The overall utilities of the terminal landing sites selected by the planning algorithms is shown in Fig. 8. The 'Global variability' plot is a distribution of the utilities of all the landing site on the map.

For both environments, random and MCTS select landing sites which on average have higher overall utility than the global variability of sites. This suggests that incorporating observations gathered during spacecraft descent helps select better sites. The average utility of sites selected by our approach was approximately 2.1. As seen in the global variability plot, sites with such a high utility were quite rare on the map. This suggests our planner was highly selective in determining the final landing site.

The median utility of the terminal sites selected by our active perception approach is $50\% - 85\%$ higher than a random selection strategy and 20% higher than a greedy strategy. As orbital data increases in noise, we can expect even larger improvements over the greedy strategy.

VII. Conclusions and Future Work

This paper presented an active perception approach to EDL which enables the lander to autonomously plan informative descent trajectories to acquire high quality sensing data during descent. The additional information gained allows the lander to select higher utility landing sites than traditional approaches to EDL. Our framework of BNs and MCTS allowed us to seamlessly combine noisy observations from multiple sensing modalities, spatial and temporal scales and plan long horizon paths while remaining robust to uncertainty. The framework is also anytime and recursive which means an history of observations does not need to be tracked, making the approach suitable for spacecraft with low onboard computational power. Results in simulated environments show promise that our approach has strong potential to improve scientific return of not only icy moon missions but EDL systems in general.

In future work, we aim to use real terrain maps generated from NASA's flyover missions, incorporate more realistic sensor noise models and dynamic constraints in our approach. We also aim to conduct hardware experiments with UAVs over ice analogue environments on Earth.

Acknowledgments

This project was funded by the ICICLES project as part of NASA's COLDTech program and by the Australian Centre for Field Robotics, The University of Sydney.

References

[1] Europa Lander Mission Concept Team. *Europa Lander Mission* Europa Lander Study 2016 Report, JPL D-97667. NASA, 2016.

[2] Europa Study Team. *Europa Lander Mission* Europa Study 2012 Report, JPL D-71990. NASA, 2012.

[3] Europa Study Team. *Europa Orbiter Mission* Europa Study 2012 Report, JPL D-71990. NASA, 2012.

[4] Desaraju, V., Michael, N., Humenberger, M., Brockers, R., Weiss, S., and Matthies, L. *Vision-based Landing Site Evaluation and Trajectory Generation Toward Rooftop Landing.* Robotics: Science and Systems (RSS), 2014.

[5] Serrano, N. *A Bayesian Framework for Landing Site Selection during Autonomous Spacecraft Descent.* Intelligent Robots and Systems (IROS), 2006.

[6] Serrano, N. and Homayoun, S. *Landing Site Selection using Fuzzy Rule-Based Reasoning.* International Conference on Robotics and Automation (ICRA), 2007.

[7] Johnson, A. and Ivanov, T. *Analysis and Testing of a LIDAR-Based Approach to Terrain Relative Navigation for Precise Lunar Landing.* AIAA Guidance Navigation and Control Conference, 2011.

[8] Ivanov, T., Huertas, A., Carson, J.M. *Probabilistic Hazard Detection for Autonomous Safe Landing.* AIAA Guidance Navigation and Control Conference, 2013.

[9] Fitzgerald, D., Walker, R. and Campbell, D. *A vision based forced landing site selection system for an autonomous UAV.* International Conference on Intelligent Sensors, Sensor Networks and Information Processing Conference, 2005.

[10] Browne C. et. al. *A Survey of Monte Carlo Tree Search Methods* IEEE Transactions on Computational Intelligence and AI in games, 2012.

[11] Arora A., Fitch, R., and Sukkarieh, S. *An Approach to Autonomous Science by Modeling Geological Knowledge in a Bayesian Framework* International Conference on Intelligent Robots and Systems (IROS), 2017

[12] Coquelin, P., and Rmi M. *Bandit algorithms for tree search.* arXiv preprint cs/0703062, 2007

[13] Nielsen, T.D. and Finn V.J. *Bayesian networks and decision graphs.* Springer Science and Business Media, 2009.

[14] Gelly, S., Kocsis, L., Schoenauer, M., Sebag, M., Silver, D., Szepesvri, C., and Teytaud, O. *The grand challenge of computer Go: Monte Carlo tree search and extensions.* Communications of the ACM, 2012, 55(3), 106-113.